PCT/JP2004/012246

日本国特許庁
JAPAN PATENT OFFICE

19.08.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　８月２９日

出願番号
Application Number: 特願２００３－３０７５０９

[ST. 10/C]: [ＪＰ２００３－３０７５０９]

出願人
Applicant(s): 松下電器産業株式会社

２００４年　９月２４日

特許庁長官
Commissioner,
Japan Patent Office

小川洋

出証番号　出証特２００４－３０８５８６４

【書類名】　特許願
【整理番号】　2038150040
【提出日】　平成15年 8月29日
【あて先】　特許庁長官殿
【国際特許分類】　H04N　4/64
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　松田　秀治
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　藪野　寛之
【特許出願人】
　【識別番号】　000005821
　【氏名又は名称】　松下電器産業株式会社
【代理人】
　【識別番号】　100081813
　【弁理士】
　【氏名又は名称】　早瀬　憲一
　【電話番号】　06(6395)3251
【手数料の表示】
　【予納台帳番号】　013527
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9600402

【書類名】特許請求の範囲
【請求項１】
それぞれマトリクス状のデータからなる複数のセクタにより構成される対象符号列に対してシンドローム演算を行うシンドローム演算器を有し、該対象符号列に対して誤り訂正を行う誤り訂正回路による誤り訂正処理と同時に、該対象符号列に対して、前記セクタ単位で誤り検出を行う誤り検出装置であって、
前記対象符号列の誤り検出符号を演算する誤り検出符号演算回路と、
前記対象符号列が論理的に連続性のない並びで入力された際に、該符号列に連続性をもたせるスキップ演算を行う誤り検出符号スキップ演算回路と、
前記シンドローム演算と同じ時点で行う第１の誤り検出符号演算処理を制御する第１の誤り検出制御回路と、
前記誤り訂正処理後に、該誤り訂正処理より得られた誤りデータ位置及び誤りデータ数値を元に、該誤りデータ位置が示すデータに対してのみ行う第２の誤り検出符号演算処理を制御すると共に、該第２の誤り検出符号演算処理による演算結果を元に、前記第１の誤り検出符号演算処理の演算結果を更新する更新処理を制御する第２の誤り検出制御回路と、
前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路による演算結果を保持するメモリと、を備える、
ことを特徴とする誤り検出装置。
【請求項２】
請求項１に記載の誤り検出装置において、
前記誤り検出符号スキップ演算回路は、前回までに入力された対象符号列の誤り検出符号を入力とし、該誤り検出符号スキップ演算回路に予め設定されたスキップ演算を行う、
ことを特徴とする誤り検出装置。
【請求項３】
請求項１に記載の誤り検出装置において、
前記メモリは、前記対象符号列が連続的な並びで入力された際は、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路によるセクタ単位の演算結果を保持し、前記対象符号列が連続的でない並びで入力された際は、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路によるセクタ毎の途中演算結果を保持する第１のメモリと、
前記第１のメモリから送信される演算結果を、前記各セクタ毎に保持する第２のメモリと、を備える、
ことを特徴とする誤り検出装置。
【請求項４】
請求項１に記載の誤り検出装置において、
前記第１のメモリは、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路による演算結果を保持する、少なくとも２つの誤り検出符号演算結果保持部からなる、
ことを特徴とする誤り検出装置。
【請求項５】
請求項４に記載の誤り検出装置において、
前記少なくとも２つの誤り検出符号演算結果保持部のうちの１つは、前記第１の誤り検出符号演算処理による演算結果を保持する、
ことを特徴とする誤り検出装置。
【請求項６】
請求項４に記載の誤り検出装置において、
前記少なくとも２つの誤り検出符号演算結果保持部のうちの１つは、前記第２の誤り検出符号演算処理による、前記第１の誤り検出符号演算処理の演算結果を更新する差分演算結果を保持する、

ことを特徴とする誤り検出装置。
【請求項７】
請求項１に記載の誤り検出装置において、
前記誤り検出符号スキップ演算回路は、前記対象符号列が論理的に連続性のない並びで入力されたとき、該対象符号列のうち前記セクタの非最終行においては、特定のスキップ演算を行い、前記対象符号列のうち前記セクタの最終行においては、該データが存在する列位置に応じた個別のスキップ演算を行う、
ことを特徴とする誤り検出装置。
【請求項８】
それぞれマトリクス状のデータからなる複数のセクタにより構成される、論理的に連続性のない並びで入力された対象符号列に対して誤り訂正を行うと同時に、該連続性のない並びで入力される対象符号列に対して前記セクタ単位で誤り検出を行う誤り検出方法であって、
前記対象符号列に対してシンドローム演算を行うシンドローム演算ステップと、
前記シンドローム演算ステップと同時に行われる、該連続性のない並びで入力される対象符号列に対して誤り検出符号演算を行う第１の誤り検出符号演算ステップと、
前記シンドローム演算ステップにおいて得られたシンドロームに基づいて、前記対象符号列の誤りデータ位置及び誤りデータ数値を計算して誤り訂正を行う誤り訂正ステップと、
前記誤り訂正ステップにおいて得られた前記誤りデータ位置及び誤りデータ数値を元に、前記対象符号列のうちの前記誤りデータ位置に対してのみ再度誤り検出符号演算を行う第２の誤り検出符号演算ステップと、
前記第２の誤り検出符号演算ステップによる演算結果を用いて、前記第１の誤り検出符号演算ステップによる演算結果を更新する更新ステップと、を含み、
前記第１の誤り検出符号演算ステップ及び第２の誤り検出演算ステップは、前記対象符号列の誤り検出符号を演算する誤り検出符号演算ステップと、前記論理的に連続性のない並びで入力された対象符号列に連続性をもたせるスキップ演算を行う誤り検出符号スキップ演算ステップと、を含む、
ことを特徴とする誤り検出方法。
【請求項９】
請求項８に記載の誤り検出方法において、
前記誤り検出符号スキップ演算ステップは、前回までに入力された対象符号列の誤り検出符号を入力とし、予め設定されたスキップ演算を行う、
ことを特徴とする誤り検出方法。
【請求項１０】
請求項８に記載の誤り検出方法において、
前記誤り検出符号スキップ演算ステップは、前記対象符号列のうち前記セクタの非最終行においては、特定のスキップ演算を行い、前記対象符号列のうち前記セクタの最終行においては、該データが存在する列位置に応じた個別のスキップ演算を行う、
ことを特徴とする誤り検出方法。
【請求項１１】
請求項１０に記載の誤り検出方法において、
前記個別のスキップ演算は、前記データが存在する列位置のうち、特定の列位置で実行されるステップ演算の演算結果を複数回利用して行う、
ことを特徴とする誤り検出方法。
【請求項１２】
請求項８に記載の誤り検出方法において、
前記更新ステップと同時に行われる、前記対象符号列に含まれるスクランブル成分を除去するスクランブル除去ステップ、または、該対象符号列にスクランブル成分を付加するスクランブル付加ステップを含む、

ことを特徴とする誤り検出方法。

【請求項１３】

請求項１２に記載の誤り検出方法において、

前記スクランブル除去ステップは、１セクタの全データが入力された後に、該セクタの全データのスクランブル成分を一度に除去する、

ことを特徴とする誤り検出方法。

【請求項１４】

請求項１２に記載の誤り検出方法において、

前記スクランブル除去ステップは、前記対象符号列のスクランブル成分を除去するためのデータを保持するテーブルを用いて行われる、

ことを特徴とする誤り検出方法。

【請求項１５】

請求項８に記載の誤り検出方法において、

前記誤り訂正及び誤り検出を行う対象データは、光記録媒体に記録されているデータである、

ことを特徴とする誤り検出方法。

【書類名】明細書
【発明の名称】誤り検出装置及び誤り検出方法
【技術分野】
【０００１】
本発明は、誤り検出装置及び誤り検出方法に関し、特に誤り訂正と誤り検出を同時に実行するための誤り検出装置及び誤り検出方法に関する。
【背景技術】
【０００２】
デジタルデータの記録／再生を行うシステムでは、再生時もしくは記録時にデータ中に誤りが発生することがあることから、その誤りを訂正し、行った誤り訂正が正しいものかどうかを判定する誤り検出を行う必要がある。
【０００３】
そして、DVDの記録／再生においては、誤り訂正ではECC（Error Correcting Code）、誤り検出ではEDC（Error Detecting Code）と呼ばれる方法が用いられる。この誤り訂正はECCブロック単位、誤り検出はセクタ単位で実行される。
【０００４】
まず、本発明にかかる誤り検出装置の理解を容易にするために、図９～図１２を用いて、ECC復号化の概要について説明する。図９は、DVD上のECCブロックの構成を示す図であり、図１０は、図９のECCブロックのC1方向についての誤り訂正を説明する図であり、図１１は、図９のECCブロックのC2方向についての誤り訂正を説明する図であり、図１２は、DVD上のセクタの構成を示す図である。
【０００５】
なお、ここでは、光記録媒体であるDVDに記録されているデータを誤り訂正する場合を例に挙げて説明するものとし、誤り訂正を行う一単位であるECCブロックは、図９に示すように１６のセクタより構成され、その１セクタは、図１２に示すような構成を有するものとする。
【０００６】
まず、DVDに記録されているデータを誤り訂正する場合、そのDVDから読み出したECC符号化されたデータを復号し、図９に示すC1方向もしくはC2方向について誤り訂正を行う。このとき、ECC復号したデータから位置多項式及び数値多項式を生成し、それらの根を求めることにより誤りデータ位置及び誤りデータ数値を算出する。
【０００７】
このC1方向あるいはC2方向について誤り訂正を行うときに、誤り訂正能力を超える誤りが存在した場合は、図１０に示すように、その誤り訂正能力を超える符号列を訂正不能符号列とし、この訂正不能符号列に関する情報を消失位置情報として記憶しておく。
【０００８】
そして、１ECCブロック中の全ての符号列に対する、C1方向もしくはC2方向についての誤り訂正が完了した後、前回と異なる方向（C2方向もしくはC1方向）について、前記消失位置情報を用いて誤り訂正を行う。
【０００９】
このように、誤り訂正を行う際に予め誤りデータの位置が分かっている場合には、該誤りデータの位置を示す消失位置情報を利用することで、前記位置多項式及び数値多項式を生成する際に、数値多項式のみを求めれば良いことになる。その結果、誤り訂正能力を向上させることができる。
【００１０】
例えば、図１０に示すように、最初に１ECCブロック中の全ての符号列に対してC1方向について誤り訂正を行い、50、90、130、200行目が訂正不能符号列であったとする。この場合、図１１に示すように、次回の誤り訂正方向であるC2方向の誤り訂正を行うときに、前回の訂正不能符号列を示す消失位置情報をもとに50、90、130、200Byte目を消失位置情報と指定することで、C2方向の誤り訂正能力を向上させることができる。

【００１１】
以上の方法により、伝送系で発生したバースト誤りに対して訂正能力を落とすことなく誤り訂正が可能となる。
【００１２】
そして、このようなECC処理の後、従来においては、前述のようにしてECC処理が施されたデータに対して、セクタ単位でEDC演算を行っていく。
【００１３】
以下、図１２及び図１３を用いて、EDC演算の概要について説明する。図１３はEDC演算回路の構成例を示す図であり、ここでは、４Byte入力のEDC演算回路を示す。
【００１４】
以下、EDCの演算式を示す。
【数１】

$$EDC(x) = \sum_{i=31}^{0} b_i x^i = I(x) \bmod \{g(x)\}$$

$$I(x) = \sum_{i=16511}^{32} b_i x^i \quad g(x) = x^{32}+x^{31}+x^4+1$$

【００１５】
前記EDC演算式は、図１３のEDC演算回路に示されるように、32ビットのシフトレジスタで、0ビット目、4ビット目、31ビット目への入力にEXOR演算回路が配置され、それぞれ、４Byteの最上位からのビット入力と31ビット目、3ビット目と31ビット目、及び30ビット目と31ビット目のEXORが演算される。
【００１６】
以上に示したEDC演算式及びEDC演算回路から理解できるように、EDC演算はデータの記録方向（図９に示されるC1方向）に沿った線形演算である。そして、図１２に示すセクタを、データの記録方向に沿ってID領域、IEC領域、RSV領域、ユーザデータ領域、EDC領域の順序で、それぞれ４Byte毎に図１３に示すEDC演算回路に入力すると、最終の４Byteであるデータ領域を入力した後の32ビットのシフトレジスタ値がEDC演算結果となる。
【特許文献１】特開平７－２３０３８８号公報
【特許文献２】特開平５－３１５９７４号公報
【特許文献３】特開平７－５０５９５号公報
【特許文献４】特許第３２７２３１７号公報
【発明の開示】
【発明が解決しようとする課題】
【００１７】
このように、従来では一般的に、誤り訂正（ECC）を行った後に誤り検出（EDC）を行う方法が用いられているが、この方法では、誤り訂正の一単位であるECCブロックをバッファから読み出して誤り訂正を行った後、再度誤り検出を行うためにECCブロックをバッファから読み出さなければならないため、メモリバッファのバンド幅の消費、処理時間の肥大化が発生する。
【００１８】
このような課題に鑑み、バッファからの一度の読み出しでECCとEDCの同時処理を行うことを考えた場合、前述したようにEDC演算処理は、データの記録方向（図９のC1方向）に沿った線形演算であるため、図１０に示すように誤り訂正装置に対して入力されるデータに論理的な連続性があるC1方向ECCの場合は、該C1方向のECCとEDC演算の同時処理が可能であるが、図１１に示すように誤り訂正装置に対して入力されるデータに論理的な連続性がないC2方向のECCの場合は、前記EDC演算と前記C2方向のECCとを同時処理するのは半導体集積回路の実装における難度が高いという課題がある。

【００１９】
本発明は前記課題を解決するためになされたものであり、ＥＣＣブロックの誤り検出を行う対象符号列が論理的に連続的でない並び（図９のC2方向）で入力された際にも、誤り訂正と誤り検出を同時に行える誤り検出装置及び誤り検出方法を提供することを目的とするものである。
【課題を解決するための手段】
【００２０】
本発明の誤り検出装置は、それぞれマトリクス状のデータからなる複数のセクタにより構成される対象符号列に対してシンドローム演算を行うシンドローム演算器を有し、該対象符号列に対して誤り訂正を行う誤り訂正回路による誤り訂正処理と同時に、該対象符号列に対して、前記セクタ単位で誤り検出を行う誤り検出装置であって、前記対象符号列の誤り検出符号を演算する誤り検出符号演算回路と、前記対象符号列が論理的に連続性のない並びで入力された際に、該符号列に連続性をもたせるスキップ演算を行う誤り検出符号スキップ演算回路と、前記シンドローム演算と同じ時点で行う第１の誤り検出符号演算処理を制御する第１の誤り検出制御回路と、前記誤り訂正処理後に、該誤り訂正処理より得られた誤りデータ位置及び誤りデータ数値を元に、該誤りデータ位置が示すデータに対してのみ行う第２の誤り検出符号演算処理を制御すると共に、該第２の誤り検出符号演算処理による演算結果を元に、前記第１の誤り検出符号演算処理の演算結果を更新する更新処理を制御する第２の誤り検出制御回路と、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路による演算結果を保持するメモリと、を備えるものである。
【００２１】
これにより、ＥＣＣブロックの誤り検出を行う対象符号列が論理的に連続的でない並びで入力された際にも、誤り訂正と誤り検出を同時に行える。
【００２２】
さらに、本発明の誤り検出装置は、前記誤り検出符号スキップ演算回路は、前回までに入力された対象符号列の誤り検出符号を入力とし、該誤り検出符号スキップ演算回路に予め設定されたスキップ演算を行うものである。
【００２３】
これにより、簡単な処理でスキップ演算が行え、誤り訂正と誤り検出の同時実行を、半導体回路に実装可能にする。
【００２４】
さらに、本発明の誤り検出装置は、前記メモリは、前記対象符号列が連続的な並びで入力された際は、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路によるセクタ単位の演算結果を保持し、前記対象符号列が連続的でない並びで入力された際は、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路によるセクタ毎の途中演算結果を保持する第１のメモリと、前記第１のメモリから送信される演算結果を、前記各セクタ毎に保持する第２のメモリ、とを備えるものである。
【００２５】
これにより、ＥＣＣブロックの誤り検出を行う対象符号列が論理的に連続的でない並びで入力された際にも、誤り訂正と誤り検出とを同時に行うことが可能となる。
【００２６】
さらに、本発明の誤り検出装置は、前記第１のメモリは、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路による演算結果を保持する、少なくとも２つの誤り検出符号演算結果保持部からなるものである。
【００２７】
さらに、本発明の誤り検出装置は、前記少なくとも２つの誤り検出符号演算結果保持部のうちの１つは、前記第１の誤り検出符号演算処理による演算結果を保持するものである。
【００２８】
さらに、本発明の誤り検出装置は、前記少なくとも２つの誤り検出符号演算結果保持部

のうちの１つは、前記第２の誤り検出符号演算処理による、前記第１の誤り検出符号演算処理の演算結果を更新する差分演算結果を保持するものである。

【００２９】

これにより、前回入力されたデータと、次に入力されたデータ間にデータの論理的な連続性をもたない場合においても、EDC演算処理を行うことが可能となる。

【００３０】

さらに、本発明の誤り検出装置は、前記誤り検出符号スキップ演算回路は、前記対象符号列が論理的に連続性のない並びで入力されたとき、該対象符号列のうち前記セクタの非最終行においては、特定のスキップ演算を行い、前記対象符号列のうち前記セクタの最終行においては、該データが存在する列位置に応じた個別のスキップ演算を行うものである。

【００３１】

これにより、簡単な処理でスキップ演算が行え、誤り訂正と誤り検出の同時実行を、半導体回路に実装可能にする。

【００３２】

また、本発明の誤り検出方法は、それぞれマトリクス状のデータからなる複数のセクタにより構成される、論理的に連続性のない並びで入力された対象符号列に対して誤り訂正を行うと同時に、該連続性のない並びで入力される対象符号列に対して前記セクタ単位で誤り検出を行う誤り検出方法であって、前記対象符号列に対してシンドローム演算を行うシンドローム演算ステップと、前記シンドローム演算ステップと同時に行われる、該連続性のない並びで入力される対象符号列に対して誤り検出符号演算を行う第１の誤り検出符号演算ステップと、前記シンドローム演算ステップにおいて得られたシンドロームに基づいて、前記対象符号列の誤りデータ位置及び誤りデータ数値を計算して誤り訂正を行う誤り訂正ステップと、前記誤り訂正ステップにおいて得られた前記誤りデータ位置及び誤りデータ数値を元に、前記対象符号列のうちの前記誤りデータ位置に対してのみ再度誤り検出符号演算を行う第２の誤り検出符号演算ステップと、前記第２の誤り検出符号演算ステップによる演算結果を用いて、前記第１の誤り検出符号演算ステップによる演算結果を更新する更新ステップと、を含み、前記第１の誤り検出符号演算ステップ及び第２の誤り検出演算ステップは、前記対象符号列の誤り検出符号を演算する誤り検出符号演算ステップと、前記論理的に連続性のない並びで入力された対象符号列に連続性をもたせるスキップ演算を行う誤り検出符号スキップ演算ステップと、を含むものである。

【００３３】

これにより、ＥＣＣブロックの誤り検出を行う対象符号列が論理的に連続的でない並びで入力された際にも、誤り訂正と誤り検出を同時に行える。

【００３４】

さらに、本発明の誤り検出方法は、前記誤り検出符号スキップ演算ステップは、前回までに入力された対象符号列の誤り検出符号を入力とし、予め設定されたスキップ演算を行うものである。

【００３５】

これにより、簡単な処理でスキップ演算が行え、入力されるデータが論理的に連続性のない場合でも、誤り訂正と誤り検出との同時実行を、容易に実現することができる。

【００３６】

さらに、本発明の誤り検出方法は、前記誤り検出符号スキップ演算ステップは、前記対象符号列のうち前記セクタの非最終行においては、特定のスキップ演算を行い、前記対象符号列のうち前記セクタの最終行においては、該データが存在する列位置に応じた個別のスキップ演算を行うものである。

【００３７】

これにより、誤り訂正と誤り検出の同時実行を、容易に実現することができる。

【００３８】

さらに、本発明の誤り検出方法は、前記個別のスキップ演算は、前記データが存在する

列位置のうち、特定の列位置で実行されるステップ演算の演算結果を複数回利用して行うものである。

【００３９】

これにより、セクタ内の誤り検出演算処理単位列に対応した、全てのスキップ演算処理を用意することなく、別個に存在するスキップ演算を複数回再利用することができ、この結果、処理資源の消費を抑えることが可能となる。

【００４０】

さらに、本発明の誤り検出方法は、前記更新ステップと同時に行われる、前記対象符号列に含まれるスクランブル成分を除去するスクランブル除去ステップ、または、該対象符号列にスクランブル成分を付加するスクランブル付加ステップを含むものである。

【００４１】

これにより、ECC処理及びEDC処理後のデータから、スクランブル成分を除去でき、あるいは該処理後のデータにスクランブル成分を付加できる。

【００４２】

さらに、本発明の誤り検出方法は、前記スクランブル除去ステップは、１セクタの全データが入力された後に、該セクタの全データのスクランブル成分を一度に除去するものである。

【００４３】

これにより、より簡単な処理によって、ECC処理及びEDC処理後のデータから、スクランブル成分を除去でき、処理時間を削減できる。

【００４４】

さらに、本発明の誤り検出方法は、前記スクランブル除去ステップは、前記対象符号列のスクランブル成分を除去するためのデータを保持するテーブルを用いて行われるものである。

【００４５】

これにより、ECC処理及びEDC処理後のデータから、全データのスクランブル成分を一度に除去することができる。

【００４６】

さらに、本発明の誤り検出方法は、前記誤り訂正及び誤り検出を行う対象データは、光記録媒体に記録されているデータであるものである。

【発明の効果】

【００４７】

本発明の誤り検出装置によれば、それぞれマトリクス状のデータからなる複数のセクタにより構成される対象符号列に対してシンドローム演算を行うシンドローム演算器を有し、該対象符号列に対して誤り訂正を行う誤り訂正回路による誤り訂正処理と同時に、該対象符号列に対して、前記セクタ単位で誤り検出を行う誤り検出装置であって、前記対象符号列の誤り検出符号を演算する誤り検出符号演算回路と、前記対象符号列が論理的に連続性のない並びで入力された際に、該符号列に連続性をもたせるスキップ演算を行う誤り検出符号スキップ演算回路と、前記シンドローム演算と同じ時点で行う第１の誤り検出符号演算処理を制御する第１の誤り検出制御回路と、前記誤り訂正処理後に、該誤り訂正処理より得られた誤りデータ位置及び誤りデータ数値を元に、該誤りデータ位置が示すデータに対してのみ行う第２の誤り検出符号演算処理を制御すると共に、該第２の誤り検出符号演算処理による演算結果を元に、前記第１の誤り検出符号演算処理の演算結果を更新する更新処理を制御する第２の誤り検出制御回路と、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路による演算結果を保持するメモリと、を備えるようにしたので、EDC演算処理を行う際に、入力されるデータが論理的な連続性のない方向であっても、EDC演算処理を実行することが可能となる。また、これにより、ECC処理とEDC処理を同時実施が可能となるので、ECCブロックをバッファから一度読み出すだけで両処理が行え、ECC処理のためにバッファから読み出した1ECCブロックを、ECC処理後に再度読み出してEDC演算処理を行う必要がなくなり、メモリバッファのバンド幅の消費、及び処理時の肥

大化を低減することができる。

【００４８】

また、本発明の誤り検出方法によれば、それぞれマトリクス状のデータからなる複数のセクタにより構成される、論理的に連続性のない並びで入力された対象符号列に対して誤り訂正を行うと同時に、該連続性のない並びで入力される対象符号列に対して前記セクタ単位で誤り検出を行う誤り検出方法であって、前記対象符号列に対してシンドローム演算を行うシンドローム演算ステップと、前記シンドローム演算ステップと同時に行われる、該連続性のない並びで入力される対象符号列に対して誤り検出符号演算を行う第１の誤り検出符号演算ステップと、前記シンドローム演算ステップにおいて得られたシンドロームに基づいて、前記対象符号列の誤りデータ位置及び誤りデータ数値を計算して誤り訂正を行う誤り訂正ステップと、前記誤り訂正ステップにおいて得られた前記誤りデータ位置及び誤りデータ数値を元に、前記対象符号列のうちの前記誤りデータ位置に対してのみ再度誤り検出符号演算を行う第２の誤り検出符号演算ステップと、前記第２の誤り検出符号演算ステップによる演算結果を用いて、前記第１の誤り検出符号演算ステップによる演算結果を更新する更新ステップと、を含み、前記第１の誤り検出符号演算ステップ及び第２の誤り検出演算ステップは、前記対象符号列の誤り検出符号を演算する誤り検出符号演算ステップと、前記論理的に連続性のない並びで入力された対象符号列に連続性をもたせるスキップ演算を行う誤り検出符号スキップ演算ステップと、を含むようにしたので、入力されるデータが論理的な連続性のない方向のECCと同時にEDC演算処理を行うことができ、ECCブロックをバッファから一度読み出すだけで両処理が行え、メモリバッファのバンド幅の消費、及び処理時の肥大化を低減することができる。

【発明を実施するための最良の形態】

【００４９】

以下、図１～図６を用いて、本実施の形態１に係る、誤り訂正と誤り検出を同時処理する誤り検出装置、及び誤り検出方法について説明する。なお、本実施の形態１においては、光記録媒体であるDVDに記録されているデータを誤り訂正（ECC）及び誤り検出(EDC)する場合を例に挙げて説明するものとし、誤り訂正を行う一単位であるECCブロックは、図９に示すように１６のセクタより構成され、その１セクタは、図１２に示すような構成を有するものとする。

【００５０】

まず、図１を用いて、本実施の形態１にかかる、ECC処理とEDC処理の同時実行処理、及び入力されるデータ符号列のスクランブル成分除去について説明する。図１は、本実施の形態１における、ECC処理とEDC処理の同時実行におけるパイプライン処理を説明する図である。

【００５１】

まずステップＳ１０において、DVDから、誤り訂正及び誤り検出を行うべきデータ符号列をC1方向あるいはC2方向の順に入力して、ECC処理のシンドローム演算処理を行う。そして、前記ステップＳ１０と同じパイプライン処理ステージであるステップＳ２０において、前記ステップＳ１０と同じ順番で入力されるデータ符号列に対して第１のEDC演算処理を実行し、セクタ単位毎に第１の誤り検出符号演算結果を得る。この第１の誤り検出符号演算結果は、入力されたデータ符号列に誤りが含まれていない場合、正常な値となり、誤りが含まれているとき、不正な値となる。

【００５２】

そして、前記ステップＳ１０により得られるシンドローム演算結果が０となり、入力されたデータ符号列に誤りが存在しないと判定された場合、ステップＳ２０以降の誤り訂正処理は行われずに、処理を終了する。そしてこの場合、前記ステップＳ２０において得られた第１の誤り検出符号演算結果が、当該誤り検出装置１０より得られる正しいEDC演算結果として出力される。

【００５３】

一方、ステップＳ１０により得られるシンドローム演算結果が０とならず、入力された

データ符号列に誤りが存在すると判定された場合は、引き続き、前記誤り訂正回路２０において誤り訂正処理が行われ、入力された符号列より誤りデータ位置及び誤りデータ数値が得られる（ステップＳ３０）。

【００５４】

そして、ステップＳ４０においては、前記ステップＳ３０で得られた誤りデータ位置、及び誤りデータ数値を元にして、DVDから入力されるデータ符号列のうち、該得られた誤り位置が示すデータに対してのみ、第２のEDC演算処理を実行して、第２の誤り検出符号演算結果を得る。このときに得られる第２の誤り検出符号演算結果は、当該誤り検出装置１０より得られる正しいEDC演算結果と、前記ステップＳ２０において得られた前記第１の誤り検出符号演算結果との差分情報である。

【００５５】

よって、ステップＳ５０において、前記ステップＳ４０で得られた第２の誤り検出符号演算結果を用いて、前記ステップＳ２０において得られた第１の誤り検出符号演算結果を更新し、その更新後の誤り検出符号演算結果が、当該誤り検出装置１０より得られる正しいEDC演算結果として出力される。

【００５６】

そして、ステップＳ５０と同じパイプライン処理ステージであるステップＳ６０で、スクランブル成分を除去するためのデータを保持するテーブルを用いて、入力されるデータ符号列のスクランブル成分を一括で除去する処理が行われる。

【００５７】

以上のような一連の処理を行うことにより、ECC処理とEDC処理の同時実行、そして入力されるデータ符号列のスクランブル成分除去を、ECCブロックをバッファから１度だけ読み出すことによって、実行することが可能となる。

【００５８】

なお、前述の説明では、DVDに記録されているデータを再生する場合を例に挙げているため、ECC処理と同時にEDC演算した後、前記データ符号列のスクランブル成分を除去するものとして説明したが、DVDにデータを記録する場合は、記録するデータをECC処理すると同時にEDC演算した後、前記データのスクランブル成分を付加することとなる。ただし、スクランブル成分を付加する場合は、ステップS60に示すスクランブル除去のように、一括して行うことはできないため、EDC演算されたデータが出力される度に付加していく。

【００５９】

次に、図２を用いて、上述したようなECC処理とEDC処理の同時処理が実行可能な、本実施の形態１における誤り検出装置の構成について説明する。図２は、本実施の形態１にかかる誤り検出装置の構成を示す図である。

【００６０】

本実施の形態１に係る誤り検出回路１０は、前記データ符号列に対してシンドローム演算を行うシンドローム演算器２１を有する誤り訂正回路２０がECC処理を実施するのと同時に、EDC処理を実施可能なものであって、図１に示すように、誤り検出符号演算回路１１１と誤り検出符号スキップ演算回路１１２とからなる演算回路１１と、第１の誤り検出符号演算結果保持部１３１と第２の誤り検出符号演算結果保持部１３２とからなる第１のメモリ１３と、第２のメモリ１２と、第１の誤り検出制御回路１４１と第２の誤り検出制御回路１４２とからなる制御回路１４と、で構成されるものである。

【００６１】

以下、各回路について説明すると、前記制御回路１４は、当該誤り検出回路１０に入力されるデータ符号列に対して誤り検出を行う際に、当該誤り検出回路１０を制御するものであり、前記第１の誤り検出制御回路１４１は、前記誤り訂正回路２０内のシンドローム演算器２１における前記データ符号列に対するシンドローム演算と同時に行われる第１のEDC演算処理（図１のステップS20）を制御し、前記第２の誤り検出制御回路１４２は、前記誤り訂正回路２０による誤り訂正処理後の第２のEDC演算処理、及び前記第１のEDC演算結果の更新処理（図１のステップS40,S50）を制御するものである。

【００６２】
そして、前記演算回路１１は、前記制御回路１４の制御の下、入力される前記データ符号列に対して誤り検出演算を行うものであり、前記誤り検出符号演算回路１１１は、前記DVDから入力されるデータ符号列の誤り検出符号を任意のデータ毎に（ここでは４バイト毎）検出するものであり、前記誤り検出符号スキップ演算回路１１２は、図９のC2方向に前記データ符号列が入力された場合に、データに連続性をもたせるためのスキップ演算を行うものである。
【００６３】
そして、前記第１のメモリ１３は、前記演算回路１１において得られた途中演算結果を保持していくものであり、前記第１の誤り検出符号演算結果保持部１３１は、前記第１の誤り検出制御回路１４１の制御の下で行われる第１のEDC演算（図１のステップS20）で得られた演算結果を保持するものであり、前記第２の誤り検出符号演算結果保持部１３２は、前記第２の誤り検出制御回路１４２の制御の下で行われる第２のEDC演算（図１のステップS40）で得られる演算結果、つまり前記第１のEDC演算処理による演算結果を更新するための差分情報を保持するものである。
【００６４】
そして、前記第２のメモリ１２は、前記第１のメモリ１３から出力されるデータをセクタ毎に保持するものであり、前記第２のEDC演算処理後には、前第２のEDC演算結果により第１のEDC演算結果が更新され、該第２のメモリ１２には、正しいEDC演算結果が保持される。
【００６５】
以上のような構成を有する本実施の形態１における誤り検出装置１０により、ECC処理とEDC処理の同時実行処理の一連の動作について、図３～図６を用いて説明する。図３は、本実施の形態１に係る、C2方向訂正時に第０セクタにおいてデータ入力単位を４Byteとしたときの誤り訂正回路、及び誤り検出回路に対するデータ入力順序を示す図であり、図４は、本実施の形態１における誤り検出回路において、C2方向訂正時の第１のEDC演算処理の一連の流れを示すフローチャート図であり、図５は、C2方向訂正時における、第１のEDC演算処理時のデータの流れを示す図であり、図６は、C2方向訂正時における、第２のEDC演算処理時のデータの流れを示す図である。なお、前述と同様、誤り訂正回路２０及び誤り検出回路１０に入力される1ECCブロックは図９に示されるような構成を有し、またセクタは図１２に示される構成を有するものとして説明する。
【００６６】
以下、図９におけるC1方向あるいはC2方向のECC処理のうち、誤り検出を行う対象符号列が連続的でないC2方向の並びでECC処理が行われる場合における、誤り検出回路１０による一連のEDC処理動作について説明する。
【００６７】
まず、誤り訂正回路２０及び誤り検出回路１０に対して入力されるデータ符号列は、図３に示すようにECCブロックのC2方向に沿って入力される。ここでは、データの入力単位及びEDC演算処理の単位が４Byteであるとする。
【００６８】
そして、この入力されたデータ符号列は、誤り訂正回路２０内のシンドローム演算器２１にて、シンドローム演算が実行されると同時に（図１のステップS10）、本誤り検出回路１０内の演算回路１１にて、第１の誤り検出制御回路１４１の制御の下、第１のEDC演算処理が行われる。以下、その詳細な処理について、図４及び図５を参照しながら説明する。なお、図９に示す１ECCブロックには、16のセクタ（第1セクタ～第15セクタ）が含まれているが、各セクタにおいて処理方法は同様であり、ここでは説明を簡略化するために、16セクタのうち第0セクタのみを例に挙げて説明する。
【００６９】
図４に示すように、第１のEDC演算処理が開始されると、第１の誤り検出制御回路１４１内に保持されているセクタ内行数カウンタ“ROW”と、セクタ内列数カウンタ“COL”を

初期化する（ステップS201,S202）。そして、第１のメモリ１３内の第１誤り検出符号演算結果保持部１３１の値（以下、「“TMP_EDC_VAL_1”」と称す。）、及び第２のメモリ１２の値（以下、「“EDC_VAL”」と称す。）を初期化する（ステップS203,S204）。

【００７０】

この後、誤り検出回路１０及び誤り訂正回路２０において、前記データ符号列を４Byte受信する。この受信したデータは、図３に示す“0”番目のデータである（ステップS205）。

【００７１】

そして、受信したデータに対して、誤り検出符号演算回路１１１において、後述する４Byte入力のEDC演算処理を行い（ステップS206）、その演算結果と、“TMP_EDC_VAL_1”のEXORをとった値を、再度“TMP_EDC_VAL_1”に保持する（ステップS207）。

【００７２】

ここで、“0”番目のデータの演算が終了後、次に入力されるデータ符号列は、図３に示す“1”番目のデータであり、先に入力された“0”番目のデータとの論理的な連続性をもたない。

【００７３】

そこで、ステップS208において、現在処理中のセクタ内行数カウンタ“ROW”を判定し、該セクタ内行数カウンタ“ROW”が、セクタ内最終行である“11”でなければ、“0”番目のデータと“1”番目のデータ間の論理的な連続性を補正するために、前記第１の誤り制御回路１４１の制御の下、前記演算回路１１内の誤り検出符号スキップ演算回路１１２によって、168Byteスキップ EDCスキップ演算処理を行う（ステップS209）。この処理の詳細については後述する。そして、この誤り検出符号スキップ演算回路１１２により得たスキップ演算結果と、前記“TMP_EDC_VAL_1”のEXORをとった値を、再度“TMP_EDC_VAL_1”に保持する（ステップS210）。この後、前記第１の誤り訂正検出回路１４１は、セクタ内行数カウンタ“ROW”を１インクリメントし（ステップS211）、次行のEDC演算処理に移行する。これらの処理を、前記セクタ内行数カウンタ“ROW”がセクタ内最終行である“11”になるまで繰り返す（ステップS205～S211）。

【００７４】

そして、ステップS208において、前記セクタ内行数カウンタ“ROW”が“11”になった場合は、図３に示す“11”番目もしくは“219”番目、・・・、“8539”番目のデータ入力に対応した４Byte入力のEDC演算処理が終了している状態となる。そして、前記セクタ内列数カウンタ“COL”が“42”になるまでは、前記セクタ内行数カウンタ“ROW”が“11”になる度に、前記誤り検出符号スキップ演算回路１１２においてその次に行うべきEDCスキップ演算処理は、前記第１の誤り検出制御回路１４１により、現在のセクタ内列数カウンタ“COL”の値に対応した個別の処理となるよう制御される（ステップS213）。

【００７５】

例えば、ステップＳ213の時点で、現在前記セクタ内列数カウンタ“COL”が“1”、つまり、図３に示す“219”番目のデータまでEDC演算処理が終了している状態であれば、該“219”番目のデータと、図３に示すセクタの最終データでありEDC領域である“8747”番目のデータまでの差分は164Byteであるため、前記第１の誤り検出制御回路１４１は、前記誤り検出符号スキップ演算回路１１２において、164Byteスキップさせる EDCスキップ演算処理を行うように制御する。また、前記ステップＳ２１３に時点で、前記セクタ内列数カウンタ“COL”が“41”、つまり図３に示す“8539”番目のデータまでのEDC演算処理が終了している状態であれば、該“8539”番目のデータと、図３に示す最終データでありEDC領域である“8747”番目のデータまでの差分は4Byteであるため、前記第１の誤り検出制御回路１４１は、前記誤り検出符号スキップ演算回路１１２において、4Byteスキップさせる EDCスキップ演算処理を行うように制御する。

【００７６】

そして、前記セクタ内列数カウンタ“COL”値毎に異なる、前記誤り検出符号スキップ演算回路１１２によるスキップ演算結果と、“TMP_EDC_VAL_1”とのEXORをとった値を、

再度“TMP_EDC_VAL_1”に保持する（ステップS214）。

【００７７】

この時点で、第１のメモリ１３内の、第１の誤り検出符号演算結果保持部１３１（“TMP_EDC_VAL_1”）には、セクタ内１列分の演算結果が保持されたこととなるので、図５に示すように、第１の誤り制御回路１４１により、この保持された演算値を第２のメモリ１２（“EDC_VAL”）に出力して、該第２のメモリ１２内の対応するセクタレジスタに出力して保持する。つまり、“EDC_VAL”と“TMP_EDC_VAL_1”とのEXORをとった値を、再度“EDC_VAL”に保持する（ステップS215）。例えば、現在第0セクタ内の“COL”が“1”、つまり図３に示す“219”番目のデータまでのEDC演算が終了し、“TMP_EDC_VAL_1”には、図３の“208”～“219”番目のデータまでのEXORをとった値が保持されている場合、図５に示すように、この値を“EDC_VAL”内の第0セクタレジスタに出力する。

【００７８】

この後、“ROW”を初期化し（ステップS216）、“COL”を1インクリメントする（ステップS217）。

【００７９】

一方、“ROW”が“11”で、“COL”が“42”である場合は、図３に示す“8747”番目のデータ入力に対応した４Byte入力EDC演算処理が終了した状態であるため、“EDC_VAL”と“TMP_EDC_VAL_1”とのEXORをとった値を再度“EDC_VAL”に保持し（ステップS218）、“ROW”及び“COL”を初期化し（ステップS219、ステップS220）、処理を終了する。

【００８０】

以上のステップにより、図３に示すECCブロック内第0セクタのEDC演算結果が全て求まり、処理を終了する。

【００８１】

そして、全てのセクタに対する第１のEDC処理が終了した後、前記誤り訂正回路２０におけるシンドローム演算値が０でなく、誤りが含まれていると判断された場合は、誤り訂正処理が施される（図１のステップS30）。そして、この誤り訂正処理において得られた誤りデータ位置及び誤りデータ数値を用いて、該誤りデータ位置が示す箇所のみの第２のEDC演算処理が行われる。

【００８２】

この第２のEDC演算処理については、第２の誤り訂正制御回路１４２の制御の下、第１のEDC演算処理と同様、演算回路１１内の誤り検出符号演算回路１１１及び誤り検出符号スキップ演算回路１１２によって、入力される前記データ符号列のうち、前記誤り訂正回路２０において得られた誤りデータ位置が示すデータについてのみ、再度EDC演算処理を行うものであり、この演算結果は、図６に示すように、第１のメモリ１３内の第２の誤り検出符号演算結果保持部１３２（以下、“TMP_EDC_VAL_2”と称す。）に保持される。この“TMP_EDC_VAL_2”に保持される値は、前述したように、当該誤り検出装置１０より得られる正しいEDC演算結果と、第１のEDC演算処理により得られ、現在“EDC_VAL”に保持されている第１の誤り検出符号演算結果との差分情報となる。

【００８３】

従って、前記誤り訂正回路２０において得られた前記誤りデータ位置に対して第２のEDC処理が終了後、前記第２の誤り検出制御回路１４２は、図６に示すように、前記“TMP_EDC_VAL_2”と、前記“EDC_VAL”のEXORをとることによって、前記“EDC_VAL”の値を更新し、正しい誤り検出符号演算結果を得る。

【００８４】

ここで、本誤り検出回路１０において誤り検出符号演算処理を行う際に、２つのメモリ、“TMP_EDC_VAL”（第１のメモリ１３）及び“EDC_VAL”（第２のメモリ１２）を使用する理由を以下に示す。

【００８５】

図３に示すC2方向のECC処理における第0セクタの“COL”が“1”の列について、データ入力順序は、0番目、1番目、2番目、・・・、11番目以降も、第2セクタである12番目、13

番目、・・・のように引き続いて入力され、図９に示すECCブロック最終セクタである第15セクタ最終行の191番目以降、C2パリティデータである最終行の207番目、続いて１インクリメントされた“COL”について、図３に示す208番目、209番目、・・・といった順序となる。従って、入力されたデータについて、前回までに入力されたデータ符号列の誤り検出符号を元にEDCスキップ演算を行った結果を保持するメモリは、第0セクタから第15セクタまでの全ての位置においても同様に使用されるため、第0セクタから第15セクタまでの任意のセクタにおけるEDC演算結果を保持するメモリが、各セクタ毎で個別に使用される必要がある。よって、本実施の形態１では、第１のメモリ１３において各セクタ内の１列分のEDC演算結果を、EXORをとることによって得、この得た第１のメモリ１３の値を、第２のメモリ１２内に設けられた各セクタ毎のレジスタに保持して、前記第１のメモリ１３内の値をリセットし、該第１のメモリ１３を次のセクタの１列分のEDC演算結果を得るために使用することができるようにしたものである。

【００８６】

次に、前述した誤り検出符号演算回路１１１におけるEDC演算処理、及び誤り検出符号ステップ演算回路１１２におけるEDCスキップ演算処理の詳細について、図１３の４Byte入力EDC演算回路を用いて説明する。

【００８７】

図１３に示すように、EDC演算回路は32ビットのシフトレジスタで構成されるため、32ビット全てに0が存在するEDC演算回路に対して、任意のビット値をEDC演算回路に入力した後のEDC演算結果である32ビットのシフトレジスタ値は、32ビット中の各ビットについて、任意のビット値を入力する前の各ビット間の関係式で表現することができる。

【００８８】

例えば、0の値である1ビットが図１３のEDC演算回路に入力された場合、32ビット中の各ビット値は、0の値である１ビットが入力される前の32ビットそれぞれの状態をBit’として以下のように示すことができる。

【００８９】

Bit[0] = Bit’ [31]
Bit[1] = Bit’ [0]
Bit[2] = Bit’ [1]
Bit[3] = Bit’ [2]
Bit[4] = Bit’ [3] ^ Bit’ [31]
Bit[5] = Bit’ [4]
・
・
Bit[30] = Bit’ [29]
Bit[31] = Bit’ [30] ^ Bit’ [31]

【００９０】

前述した1ビットスキップ演算処理式は、前回にEDC演算回路に対して入力されたデータと、新たにEDC演算回路に対して入力されたデータ間の差分が１ビットであるときに用いる、演算回路１１内の誤り検出符号演算回路１１１の演算処理式に他ならない。

【００９１】

さらに、図４に示すステップＳ２０９他で、データの連続性をたもつために、データを168Byteスキップさせる EDCスキップ演算処理を考えるとき、32ビット全てに0が存在する図１３で示すEDC演算回路に対して、全て0の値である1344ビット（168Byte）が入力された後のEDC演算結果である32ビットのシフトレジスタ値は、0の値である1ビットが入力される前の32ビットそれぞれの状態をBit’として以下のように示すことができる。

【００９２】

Bit[0] = Bit’ [0] ^ Bit’ [1] ^ Bit’ [2] ^ Bit’ [3] ^ Bit’ [5] ^ Bit’ [6] ^ Bit’ [7] ^ Bit’ [9] ^ Bit’ [10] ^ Bit’ [11] ^ Bit’ [13] ^ Bit’ [14] ^ Bit’ [15] ^ Bit’ [17] ^ Bit’ [18] ^ Bit’ [19] ^ Bit’ [21] ^ Bit’ [22] ^Bit’ [23] ^Bit

```
' [24] ^ Bit' [25] ^ Bit' [26] ^ Bit' [27] ^ Bit' [28] ^ Bit' [29]^Bit' [30] ^ B
it' [31]
Bit[ 1] = Bit' [ 0] ^ Bit' [ 1] ^ Bit' [ 2] ^ Bit' [ 3] ^ Bit' [ 4] ^ Bit' [ 6]
^ Bit' [ 7] ^ Bit' [ 8] ^ Bit' [10] ^ Bit' [11] ^ Bit' [12] ^ Bit' [14]  ^ Bit'
[15] ^ Bit' [16] ^ Bit' [18] ^ Bit' [19] ^ Bit' [20] ^ Bit' [22] ^ Bit' [23] ^ B
it' [24] ^ Bit' [25] ^ Bit' [26] ^ Bit' [27] ^ Bit' [28] ^ Bit' [29] ^ Bit' [30]
 ^ Bit' [31]
Bit[ 2] = Bit'[ 0] ^ Bit' [ 1] ^ Bit' [ 2] ^ Bit' [ 3] ^ Bit' [ 4] ^ Bit' [ 5] ^
 Bit' [ 7] ^ Bit' [ 8] ^ Bit' [ 9] ^ Bit' [11] ^ Bit' [12] ^ Bit' [13] ^ it' [15
] ^ Bit' [16] ^ Bit' [17] ^ Bit' [19] ^ Bit' [20] ^ Bit' [21] ^ Bit' [23] ^  Bit
' [24] ^ Bit' [25] ^ Bit' [26] ^ Bit' [27] ^ Bit' [28] ^ Bit' [29] ^Bit' [30] ^
Bit' [31]
Bit[ 3] = Bit' [ 0] ^ Bit' [ 1] ^ Bit' [ 2] ^ Bit' [ 3] ^ Bit' [ 4] ^ Bit' [ 5]
^ Bit' [ 6] ^ Bit' [ 8] ^ Bit' [ 9] ^ Bit' [10] ^ Bit' [12] ^ Bit' [13] ^Bit' [1
4] ^ Bit' [16] ^ Bit' [17] ^ Bit' [18] ^ Bit' [20] ^ Bit' [21] ^Bit' [22] ^ Bit
' [24] ^ Bit' [25] ^ Bit' [26] ^ Bit' [27] ^ Bit' [28] ^  Bit' [29] ^ Bit' [30]
^ Bit' [31]
Bit[ 4] = Bit' [ 4] ^ Bit' [24]
Bit[ 5] = Bit' [ 5] ^ Bit' [25]
 .
 .
Bit[30] = Bit' [ 2] ^ Bit' [ 3] ^ Bit' [ 6] ^ Bit' [ 7] ^ Bit' [10] ^ Bit' [11]
^ Bit' [14] ^ Bit' [15] ^ Bit' [18] ^ Bit' [19] ^ Bit' [30]
Bit[31] = Bit' [ 0] ^ Bit' [ 1] ^ Bit' [ 2] ^ Bit' [ 4] ^ Bit' [ 5] ^ Bit' [ 6]
^ Bit' [ 8] ^ Bit' [ 9] ^ Bit' [10] ^ Bit' [12] ^ Bit' [13] ^ Bit' [14] ^Bit' [1
6] ^ Bit' [17] ^ Bit' [18] ^ Bit' [20] ^ Bit' [21] ^ Bit' [22] ^ Bit' [23] ^ Bit
' [24] ^ Bit' [25] ^ Bit' [26] ^ Bit' [27] ^ Bit' [28] ^ Bit' [29] ^ Bit' [30]
```

【００９３】

前述した168Byteスキップ演算処理式は、前回にEDC演算回路に対して入力されたデータと、新たにEDC演算回路に対して入力されたデータ間の差分が168Byteであるときに用いる、演算回路１１内の誤り検出符号スキップ演算回路１１２の１つの演算処理式に他ならない。そして、図４のステップＳ２１３においては、各“COL”値に応じたスキップ演算処理を行う必要があり、この各“COL”値に応じたスキップ演算処理式ついても、前記168Byteスキップ演算処理式と同様にして得ることができ、図４におけるステップＳ２１３では、各“COL”値に応じたスキップ演算処理式を４２通り用意すればよいこととなる。

【００９４】

なお、前述したように図４におけるステップＳ２１３の処理のために、“COL”値に応じたスキップ演算処理式を全部で４２通り用意するのではなく、前記スキップ演算を分割して処理するようにすれば、前記ステップＳ２１３におけるステップ数を削減することが可能となる。

【００９５】

以下、スキップ演算処理を分割処理により行う場合について、図７の168ByteスキップEDC補正演算処理説明図、及び図８の168ByteスキップEDC補正演算処理の分割処理説明図を用いて説明する。

【００９６】

図７に示すように、図３に示す“COL”が“0”の場合は、“ROW”が“11”の時点で168Byteスキップさせる EDCスキップ演算を行う必要がある。このように、セクタ中の全てのEDC演算を行うためには、EDCスキップ演算処理は“COL”が“0”から“41”までの42通りについて用意しておく必要があり、処理資源を大きく消費してしまう。従って、例えば、データを168Byteスキップさせる EDCスキップ演算処理を行うことと、42Byteスキップさせ

るEDCスキップ演算処理を４回行うことが等価になるので、このことを利用して、データを168Byteスキップさせる EDCスキップ演算処理を個別に用意するのではなく、別個に存在する、データを42Byteスキップさせる EDCスキップ演算処理を４回再利用する方法に変更する。

【００９７】

以上のように、本実施の形態１によれば、入力されたデータ符号列に対して、入力データ単位Byteに対応した第１のEDC演算処理を行い、その演算結果を第１のメモリ１３内の第１の誤り検出符号演算結果保持部１３１に保持した後に、該第１の誤り検出符号演算結果保持部１３１の値と、次に入力されるデータまでの差分Byte数のEDCスキップ演算処理の演算結果のEXORをとることを繰り返すようにしたので、EDC演算処理を行う際に、入力されるデータが論理的な連続性のない方向（図９のC2方向）であっても、EDC演算処理を実行することが可能となり、前記ECC及びEDCの両処理の半導体集積回路の実装を実現できる。また、これにより、ECC処理とEDC処理を同時実施が可能となるので、ECCブロックをバッファから一度読み出すだけで両処理が行え、ECC処理のためにバッファから読み出した1ECCブロックを、ECC処理後に再度読み出してEDC演算処理を行う必要がなくなり、メモリバッファのバンド幅の消費、及び処理時の肥大化を低減することができる。

【００９８】

また、前記演算回路１１内の誤り検出符号スキップ演算回路１１２における、EDCスキップ演算処理を、分割処理方法を用いて行えば、セクタ内のEDC処理単位列に対応した、全てのEDCスキップ演算処理を用意することなく、別個に存在するEDCスキップ演算処理を複数回再利用することができ、この結果、処理資源の消費を抑えることが可能となる。

【００９９】

なお、前述の説明においては、データに論理的な連続性のないC2方向のECC処理とEDC演算処理とを同時実行する場合のEDC演算処理について詳述したが、本実施の形態１にかかる誤り検出回路１０において、データに論理的な連続性があるC1方向のECC処理とEDC演算処理とを同時実行することも可能である。この場合第１のEDC演算処理は、前述したC2方向の処理とは異なりスキップ演算をする必要がないので、演算回路１１内の誤り検出符号演算回路１１１のみを使用し、演算処理単位（ここでは４Byte）毎に得られた演算結果を第1のメモリ１３内の第１の誤り検出符号演算結果保持部１３１に保持していき、１セクタのEDC演算処理が終了後、前記第１の誤り検出符号演算結果保持部１３１に保持した演算結果を、第２のメモリ１２内の対応するセクタレジスタに保持し、次のセクタのEDC演算処理に移行する処理を繰り返す。そして、第２のEDC演算処理は、前述したC2方向の処理と同様、前記誤り検出回路２０にて得られた誤りデータ位置及び誤りデータ数値を元に、入力されるデータ符号列のうち、前記誤りデータ位置のデータに対してのみ再度EDC演算処理を行い、得られた演算結果を第１のメモリ１３内の第2の誤り検出符号演算結果保持部１３２に保持し、該第２のEDC処理が終了後、その時点で第２のメモリ１２に保持されている第１のEDC演算結果を、前記第２の誤り検出符号演算結果保持部１３２に保持された演算結果を用いて更新し、正しいEDC演算結果を得る。

【産業上の利用可能性】

【０１００】

本発明の誤り検出装置及び誤り検出方法は、ECC復号化されたデジタルデータを高倍速で記録または再生する光ディスク記録再生装置において、データを記録再生するときに、誤り訂正及び誤り検出を同時処理させる際に有用である。

【図面の簡単な説明】

【０１０１】

【図１】本発明の実施の形態１における、ECC処理とEDC処理の同時実行した際のパイプライン処理を示す図である。

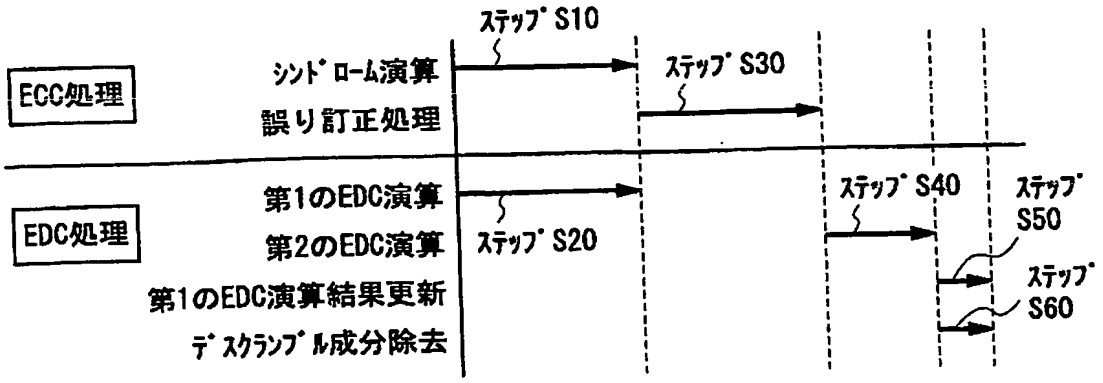
【図２】本発明の実施の形態１における、誤り検出回路の構成を示す図である。

【図３】本発明の実施の形態１における、C2方向訂正時に第0セクタにおいてデータ入力単位を４Byteとしたときの誤り訂正回路、及び誤り検出回路に対するデータ入力

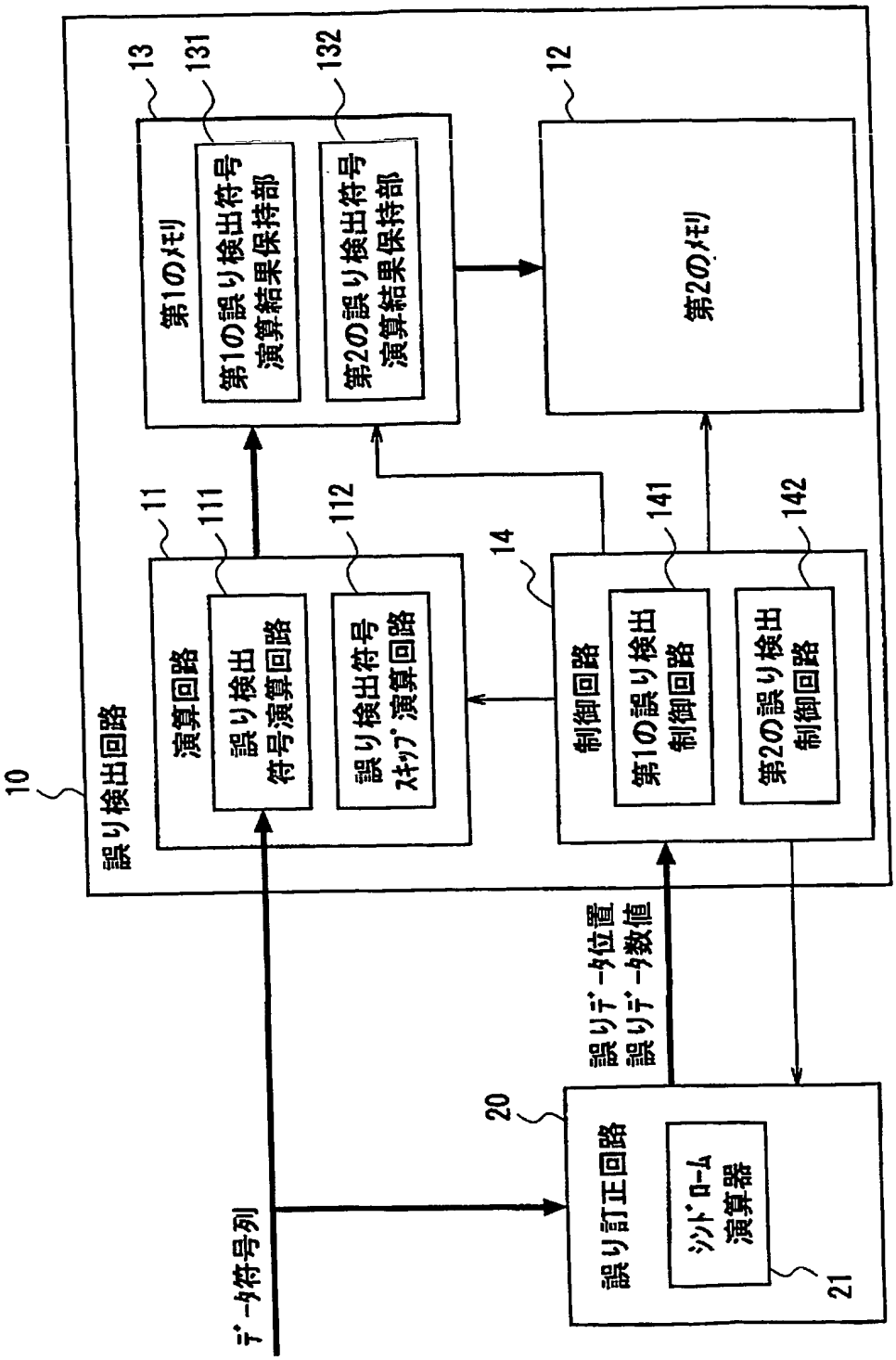
順序を示す図である。
【図４】本発明の実施の形態１における誤り検出回路において、C2方向訂正時の第１のEDC演算処理の一連の流れを示すフローチャート図である。
【図５】本発明の実施の形態１における誤り検出回路のC2方向訂正時における第１のEDC演算処理のデータの流れを示す図である。
【図６】本発明の実施の形態１における誤り検出回路の、C2方向訂正時における第２のEDC演算処理のデータの流れを示す図である。
【図７】本発明の実施の形態１における誤り検出回路の、誤り検出符号スキップ演算回路において、データを168Byteスキップさせるスキップ演算処理を示す図である。
【図８】本発明の実施の形態１における誤り検出回路の、誤り検出符号スキップ演算回路において、データを168Byteスキップさせるスキップ演算処理の分割処理を示す図である。
【図９】DVDに記録されたデータを誤り訂正単位ブロック（ECCブロック）に分けたときのECCブロック構成図である。
【図１０】ECCブロックのC1方向についての誤り訂正実施例を示す図である。
【図１１】ECCブロックのC2方向についての誤り訂正実施例を示す図である。
【図１２】ECCブロック上のセクタの構成を示す図である。
【図１３】４Byte入力のEDC演算回路を示す図である。

【符号の説明】
【０１０２】
１０　誤り検出回路
１１　演算回路
１２　第２のメモリ
１３　第１のメモリ
１４　制御回路
２０　誤り訂正回路
２１　シンドローム演算器
１１１　誤り検出符号演算回路
１１２　誤り検出符号スキップ演算回路
１３１　第１の誤り検出符号演算結果保持部
１３２　第２の誤り検出符号演算結果保持部
１４１　第１の誤り制御回路
１４２　第２の誤り制御回路

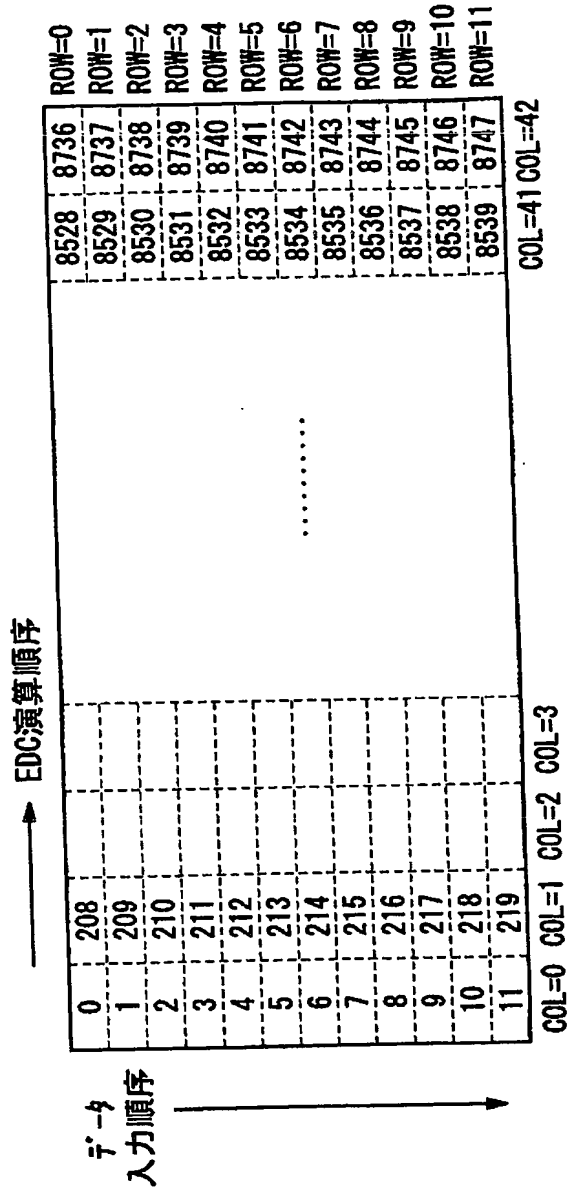
【書類名】図面
【図１】


ステップ S10
ECC処理
シンドローム演算
誤り訂正処理
ステップ S30
EDC処理
第1のEDC演算
第2のEDC演算
第1のEDC演算結果更新
デスクランブル成分除去
ステップ S20
ステップ S40
ステップ S50
ステップ S60

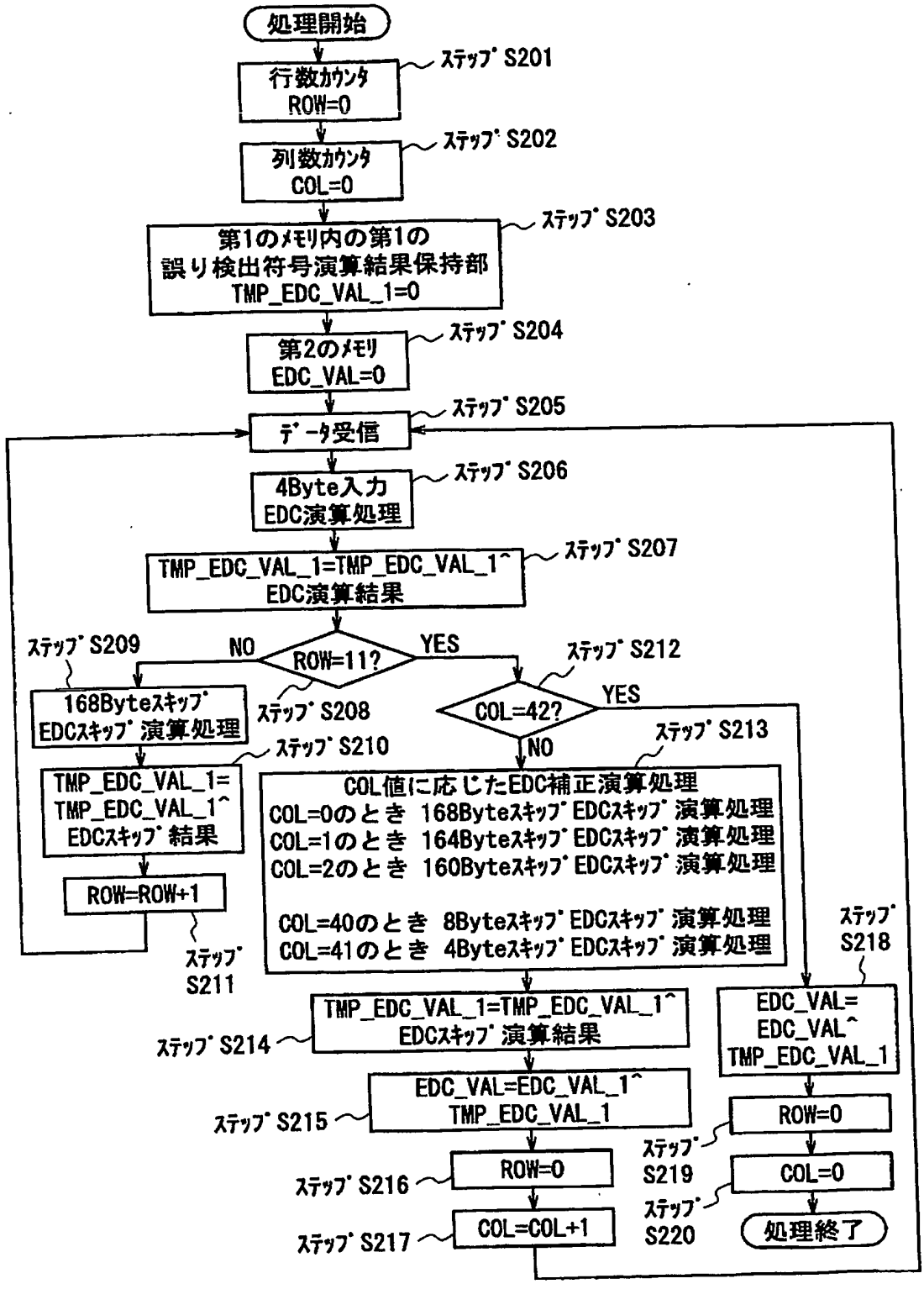
【図２】


誤り検出回路
10
演算回路
11
誤り検出
符号演算回路
111
誤り検出符号
ｼﾝﾄﾞﾛｰﾑ演算回路
112
第1のﾒﾓﾘ
13
第1の誤り検出符号
演算結果保持部
131
第2の誤り検出符号
演算結果保持部
132
第2のﾒﾓﾘ
12
制御回路
14
第1の誤り検出
制御回路
141
第2の誤り検出
制御回路
142
誤りﾃﾞｰﾀ位置
誤りﾃﾞｰﾀ数値
ﾃﾞｰﾀ符号列
誤り訂正回路
20
ｼﾝﾄﾞﾛｰﾑ
演算器
21

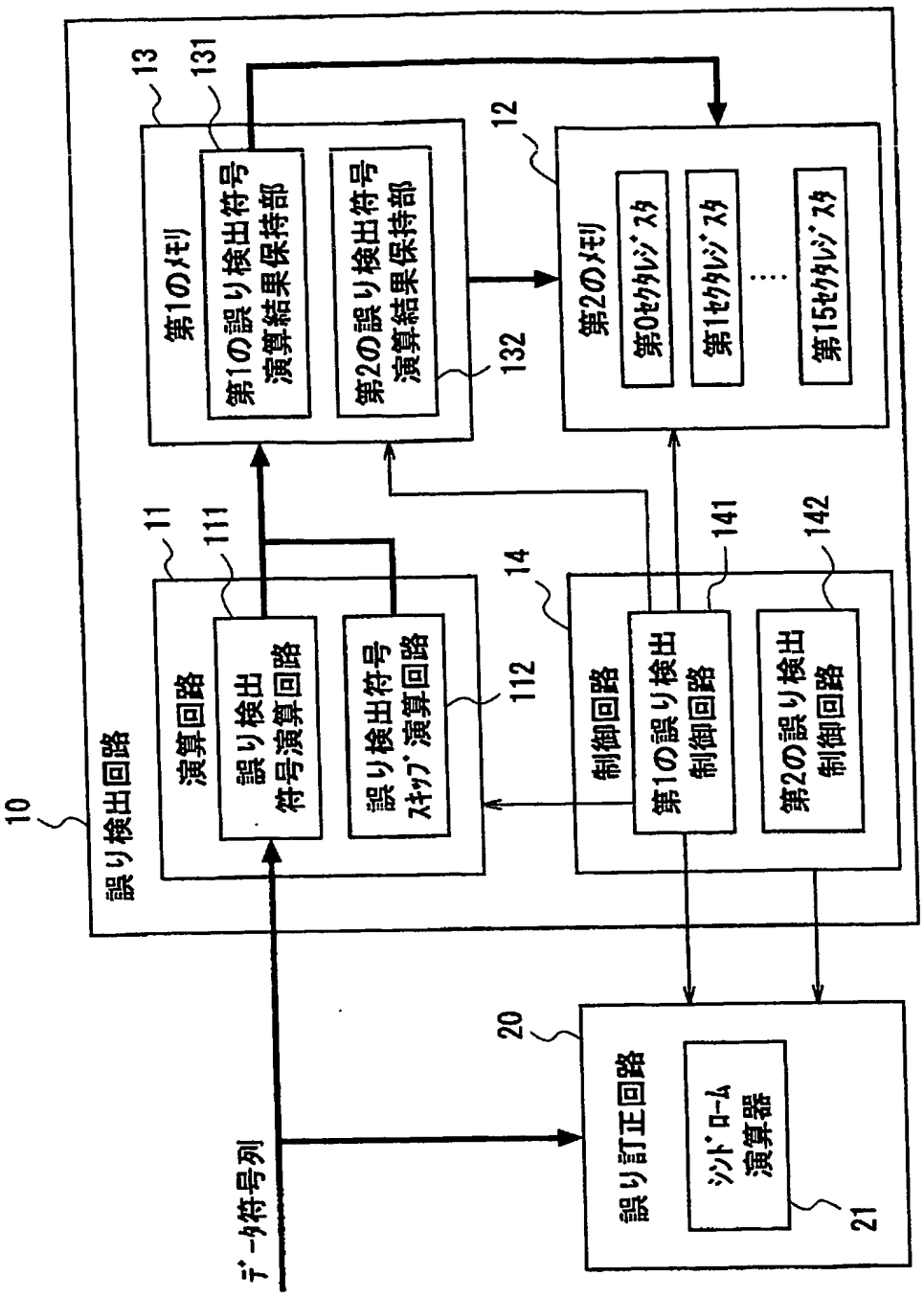
【図３】


EDC演算順序
データ入力順序
ROW=0
ROW=1
ROW=2
ROW=3
ROW=4
ROW=5
ROW=6
ROW=7
ROW=8
ROW=9
ROW=10
ROW=11
8528 8736
8529 8737
8530 8738
8531 8739
8532 8740
8533 8741
8534 8742
8535 8743
8536 8744
8537 8745
8538 8746
8539 8747
COL=41 COL=42
0 208
1 209
2 210
3 211
4 212
5 213
6 214
7 215
8 216
9 217
10 218
11 219
COL=0 COL=1 COL=2 COL=3

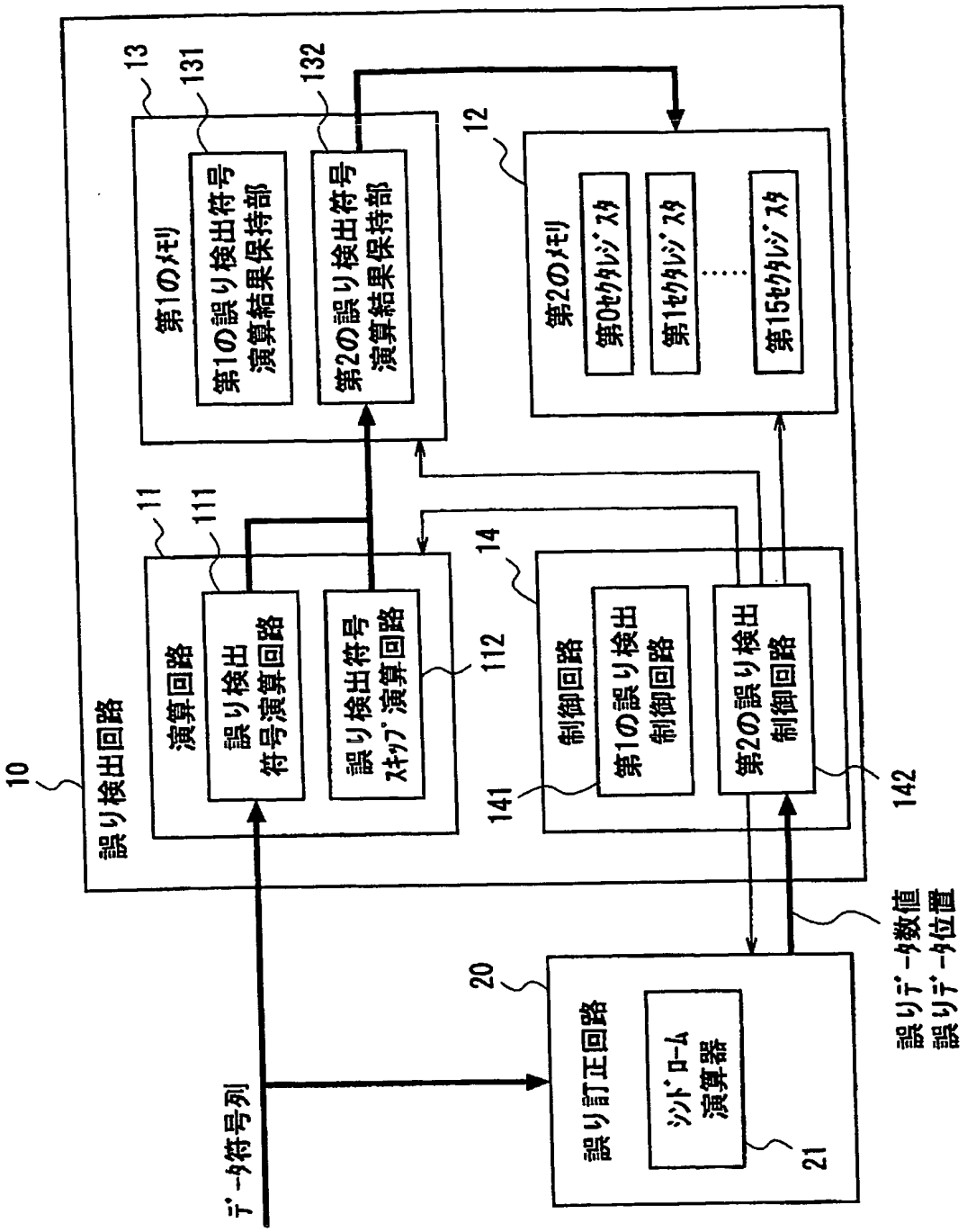
【図４】


処理開始
行数カウンタ ROW=0
ステップ S201
列数カウンタ COL=0
ステップ S202
第1のメモリ内の第1の誤り検出符号演算結果保持部 TMP_EDC_VAL_1=0
ステップ S203
第2のメモリ EDC_VAL=0
ステップ S204
データ受信
ステップ S205
4Byte入力 EDC演算処理
ステップ S206
TMP_EDC_VAL_1=TMP_EDC_VAL_1^ EDC演算結果
ステップ S207
ROW=11?
NO
YES
ステップ S208
ステップ S209
168Byteスキップ EDCスキップ 演算処理
ステップ S210
TMP_EDC_VAL_1= TMP_EDC_VAL_1^ EDCスキップ 結果
ROW=ROW+1
ステップ S211
COL=42?
YES
NO
ステップ S212
ステップ S213
COL値に応じたEDC補正演算処理
COL=0のとき 168Byteスキップ EDCスキップ 演算処理
COL=1のとき 164Byteスキップ EDCスキップ 演算処理
COL=2のとき 160Byteスキップ EDCスキップ 演算処理
COL=40のとき 8Byteスキップ EDCスキップ 演算処理
COL=41のとき 4Byteスキップ EDCスキップ 演算処理
TMP_EDC_VAL_1=TMP_EDC_VAL_1^ EDCスキップ 演算結果
ステップ S214
EDC_VAL=EDC_VAL_1^ TMP_EDC_VAL_1
ステップ S215
ROW=0
ステップ S216
COL=COL+1
ステップ S217
EDC_VAL= EDC_VAL^ TMP_EDC_VAL_1
ステップ S218
ROW=0
ステップ S219
COL=0
ステップ S220
処理終了

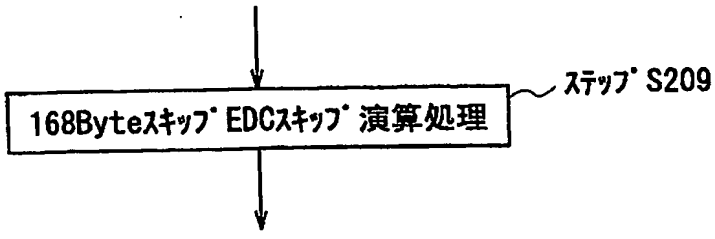
【図５】


誤り検出回路
10
演算回路
11
誤り検出
符号演算回路
111
誤り検出符号
ｼﾝﾄﾞﾛｰﾑ演算回路
112
第1のﾒﾓﾘ
13
第1の誤り検出符号
演算結果保持部
131
第2の誤り検出符号
演算結果保持部
132
第2のﾒﾓﾘ
12
第0ｾｸﾀﾚｼﾞｽﾀ
第1ｾｸﾀﾚｼﾞｽﾀ
第15ｾｸﾀﾚｼﾞｽﾀ
制御回路
14
第1の誤り検出
制御回路
141
第2の誤り検出
制御回路
142
ﾃﾞｰﾀ符号列
誤り訂正回路
20
ｼﾝﾄﾞﾛｰﾑ
演算器
21

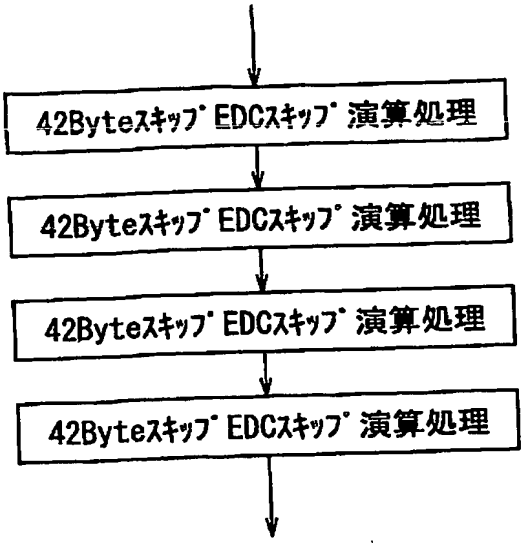
【図６】


誤り検出回路
10
演算回路
11
誤り検出
符号演算回路
111
誤り検出符号
ｽｷｯﾌﾟ演算回路
112
第1のﾒﾓﾘ
13
第1の誤り検出符号
演算結果保持部
131
第2の誤り検出符号
演算結果保持部
132
第2のﾒﾓﾘ
12
第0ｾｸﾀﾚｼﾞｽﾀ
第1ｾｸﾀﾚｼﾞｽﾀ
第15ｾｸﾀﾚｼﾞｽﾀ
制御回路
14
第1の誤り検出
制御回路
141
第2の誤り検出
制御回路
142
誤りﾃﾞｰﾀ数値
誤りﾃﾞｰﾀ位置
誤り訂正回路
20
ｼﾝﾄﾞﾛｰﾑ
演算器
21
ﾃﾞｰﾀ符号列

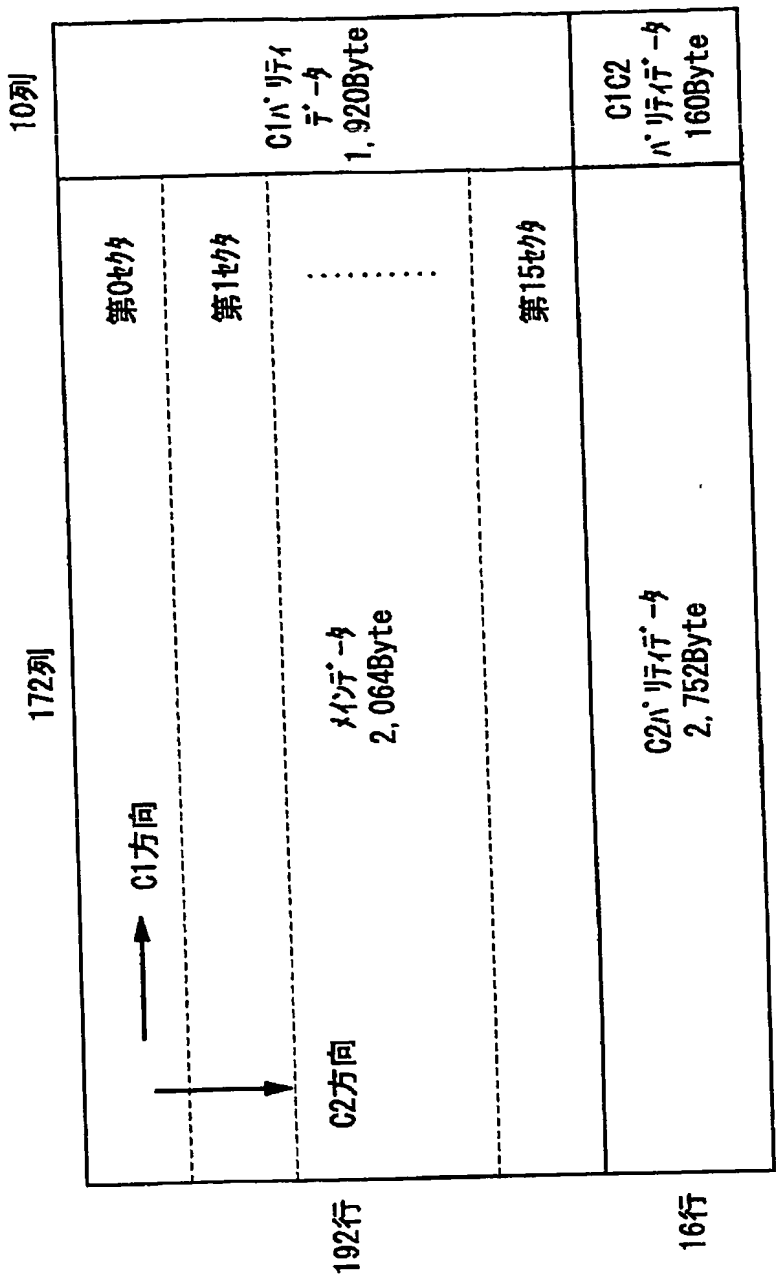
【図７】


168Byteｽｷｯﾌﾟ EDCｽｷｯﾌﾟ 演算処理
ｽﾃｯﾌﾟ S209

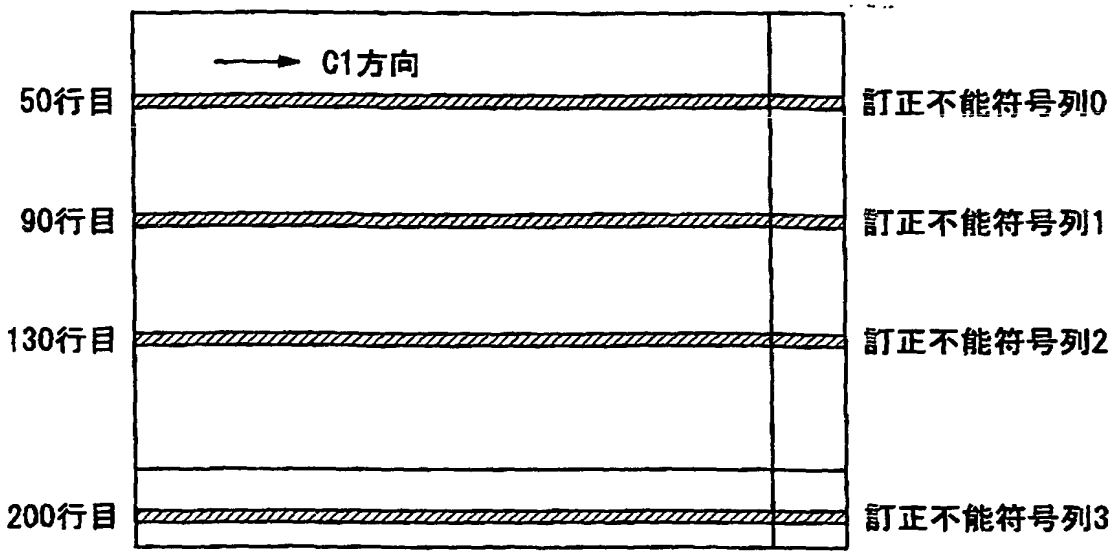
【図８】


42Byteｽｷｯﾌﾟ EDCｽｷｯﾌﾟ 演算処理
42Byteｽｷｯﾌﾟ EDCｽｷｯﾌﾟ 演算処理
42Byteｽｷｯﾌﾟ EDCｽｷｯﾌﾟ 演算処理
42Byteｽｷｯﾌﾟ EDCｽｷｯﾌﾟ 演算処理

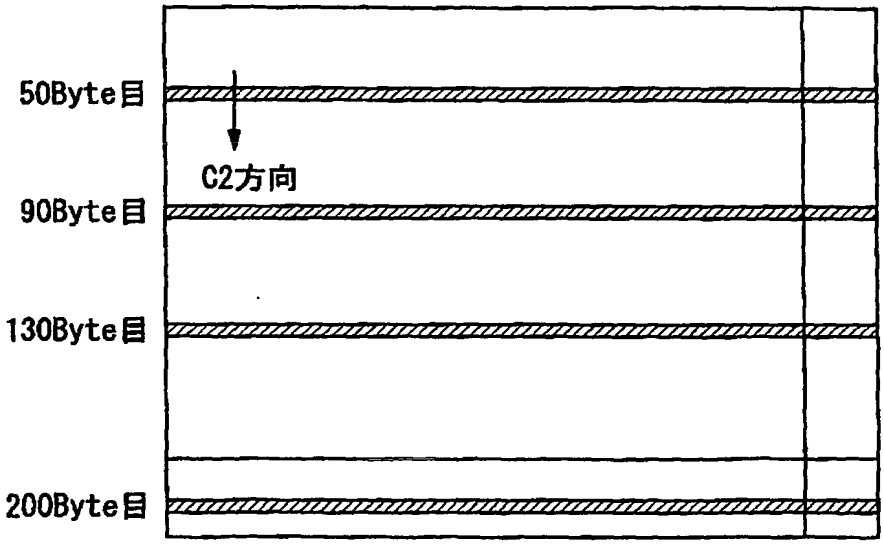
【図９】


10列
172列
第0セクタ
第1セクタ
第15セクタ
C1パリティデータ
1,920Byte
C1C2
パリティデータ
160Byte
メインデータ
2,064Byte
C2パリティデータ
2,752Byte
C1方向
C2方向
192行
16行

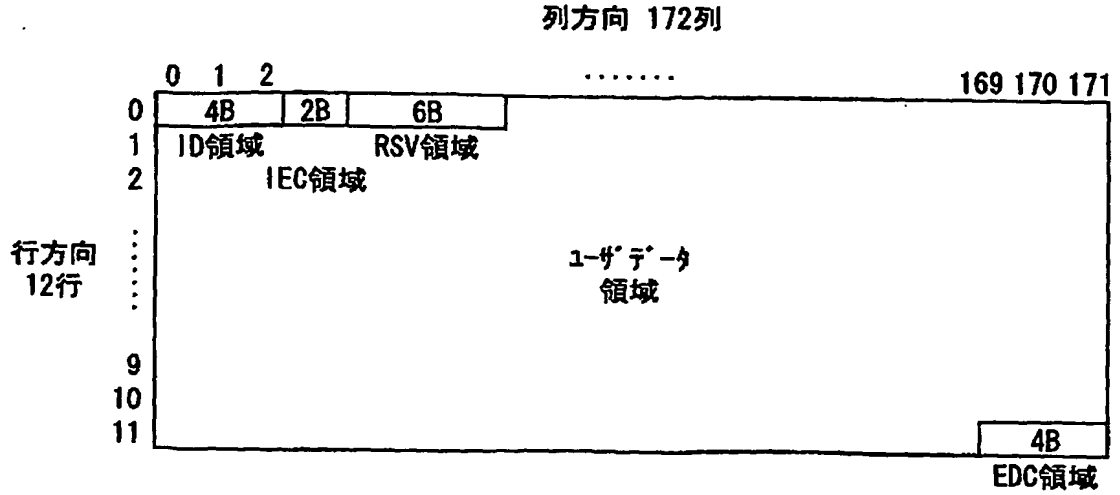
【図１０】


C1方向
50行目
90行目
130行目
200行目
訂正不能符号列0
訂正不能符号列1
訂正不能符号列2
訂正不能符号列3

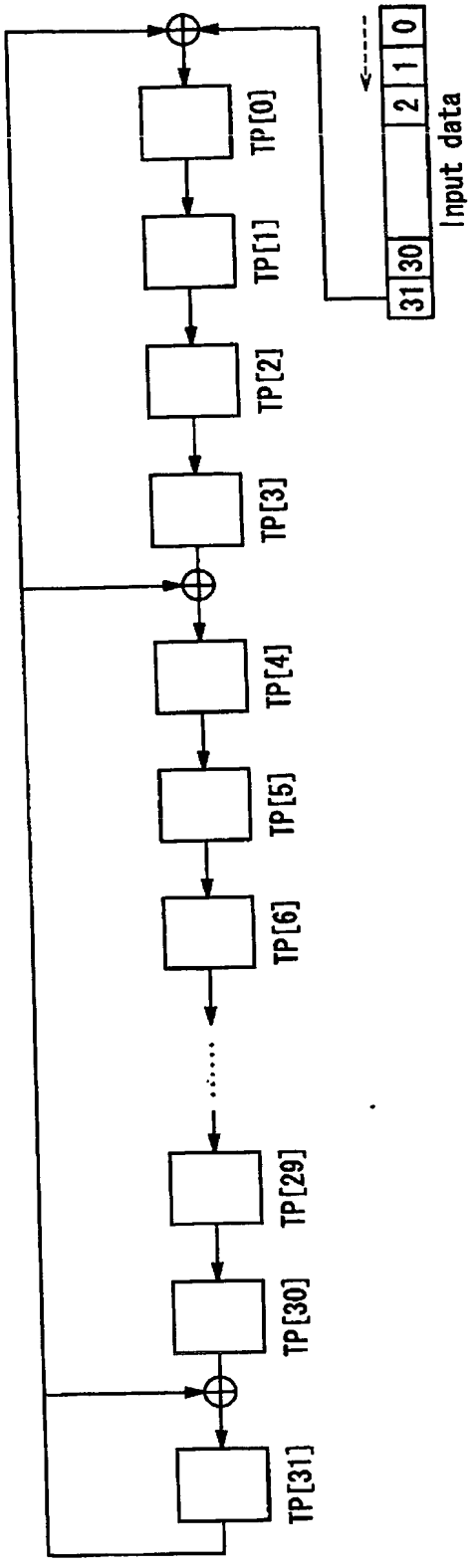
【図１１】


50Byte目
C2方向
90Byte目
130Byte目
200Byte目

【図１２】


列方向 172列
0 1 2
169 170 171
0
1
2
行方向
12行
9
10
11
4B
2B
6B
ID領域
RSV領域
IEC領域
ユーザデータ
領域
4B
EDC領域

【図１３】



【書類名】要約書
【要約】
【課題】　ECCブロックの誤り検出を行う対象符号列が、論理的に連続的でない並びで入力された際にも、誤り訂正と誤り検出を同時に行える誤り検出装置及び誤り検出方法を提供する。
【解決手段】　誤り訂正と同時に誤り検出を行う誤り検出装置であって、ECCブロックをバッファから１度だけ読み出し、誤り検出において、該誤り検出を行う対象符号列が論理的に連続的でない並びで入力された際には、該符号列に連続性をもたせるスキップ演算処理を行うことにより、論理的に連続的でない並びで入力されたECC処理とEDC演算処理とを同時に行うことを可能にする。
【選択図】　図１

特願２００３－３０７５０９

出　願　人　履　歴　情　報

識別番号　［０００００５８２１］

１．変更年月日　１９９０年　８月２８日
［変更理由］　新規登録
住　所　大阪府門真市大字門真１００６番地
氏　名　松下電器産業株式会社